USING LANGUAGE MODELS IN CAUSAL STORY GENERATION

An Undergraduate Thesis
Presented to
The Academic Faculty

By

Siyan Li

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science in the
College of Computing
Computer Science

Georgia Institute of Technology

Nov 2020

USING LANGUAGE MODELS IN CAUSAL STORY GENERATION

Faculty Readers:

Dr. Mark Riedl
School of Interactive Computing
Georgia Institute of Technology

Dr. Diyi Yang
School of Interactive Computing
Georgia Institute of Technology

Date approved: November 16th, 2020

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Story generation remains a challenge because it is still difficult to automatically generate logically coherent yet natural stories. In this thesis, we propose an approach to this problem by combining our previous pipeline for story generation and the GPT-2 language model [1]. This new architecture involves filtering generation results from GPT-2, and it outperforms the unfiltered GPT-2 model on tasks such as maintaining a single plotline and having events occurring in a sensible order.

CHAPTER 1
INTRODUCTION

One core aspect of computational creativity is automated story generation. As humans, we have passed down information from generation to generation via stories; what if machines could create stories as logically and linguistically coherent as ours? The machine must be able to distinguish references to different characters previously established in the stories, as well as recognize how different events in the stories are linked together in causal relationships, and adjust the flow of the stories accordingly. Earlier attempts at automated story generation relied on deterministic [2, 3] planning, while recent research focuses on neural text generation, which involves generating stories using neural network models from a seed sentence. Despite the various approaches to computer-generated stories, the task still poses a challenging problem.

Large-scale pre-trained transformer-based [4] language models such as ELMo [5], BERT [6], GPT-2 [1], Grover [7], and CTRL [8] have been proven capable of generating fluent passages of natural text. For instance, GPT-2 has been utilized in patent generation [9], and Zhang *et al.* [10] trained the GPT-2 model on dialogue-like exchanges on Reddit to obtain a model for dialogue continuation. However, one issue with these large-scale transformer-based model is the lack of logical coherency in generated texts. There may be drastic changes in the style of text or abrupt introductions of characters and events among other things. The reason for such tendencies lies in the nature of the model and the data the model is trained on. Language models like GPT-2 produce the next token using a probability distribution of the next token t_{n+1} given the history of tokens $t_{1..n}$, $p(t_{n+1}|t_{1..n})$. The emphasis on token history indicates that language models lack a focus on planning, hence there is often no explicit causality within these models. Furthermore, the model is trained on the WebText corpus, which consists of natural language scraped from the internet [1].

Most of these texts are not stories. Therefore, training on such corpora would not produce models that are experts in generating stories.

An alternative route to story generation is abstracting stories into sequences of discrete events [11]. The automated story generation task, by this definition, would be selecting a series of events that are likely to occur sequentially based on commonsense knowledge. This type of commonsense knowledge needs to be reinforced onto the event selection process, since events themselves do not represent commonsense causalities. Martin, Sood, and Riedl [12] proposed an agent to play tabletop role-playing games such as Dungeons and Dragons. They used the mechanics of an event generation model in addition to converting natural language to and from discretized events; the next agent event is selected by first filtering the list of generated events using rules derived from *VerbNet* semantics [13], then utilizing a Deep-Q-Network (DQN) to choose the exact action from the filtered events. Although stories generated via event selection tend to be more logically sound and compact, the pipeline's mechanism converting from events to natural language involves omitting information to create events, which reduces the variability of generated texts, rendering it less natural than stories generated by single-step models.

Considering the advantages and disadvantages of both approaches to story generation, we propose a pipeline where we remove the event-related segments from the pipeline from Martin, Sood, and Riedl [12] and directly combine neural story generation with verb semantics obtained from *VerbNet* [13]. We will be using the GPT-2 language model (117M) [1] to generate the top-k sentences based on the probability distribution over the following tokens given a seed sentence and the history of tokens. The set of sentences will then be translated into *VerbNet* frames, and the sentences containing frames that are not valid given the current story state will be eliminated from the selection. The framework would then randomly select a response from the remaining set of valid sentences. We are interested in seeing if removing events from the original pipeline and replacing it with the GPT-2 model would result in stories that are at least as logically coherent as stories generated by the

original pipeline.

The contributions of this thesis include: (1) propose an approach to prepare story datasets for language model fine-tuning; (2) improve upon previous pipelines by eliminating events from the process.

CHAPTER 2

RELATED WORK

2.1 Eventified Story Generation

A story can be defined as a sequence of different events, each event corresponding to one or more sentences in the original story. A natural language sentence can be converted into an event represented by a 4-tuple $<$ *subject, verb, direct object, modifier* $>$, where the entries correspond to the subject, verb, direct object, and modifier (e.g. causal complements) respectively [11]. Additional operations on the event to further enhance model performances are: (1) generalizing entities by identifying named entities and replacing them with the corresponding tags from the *WordNet* [14] Synsets, and (2) named entity numbering. Such abstractions have been shown to improve the performance of both event-to-event problems and event-to-sentence problems. Martin *et al.* [11] parsed each sentence with the Stanford Dependency Parser [15] to generate individual clauses to be eventified.

Martin, Sood, and Riedl [12] incorporated this abstraction into their agent to collaborate with humans in tabletop role-playing games such as Dungeons and Dragons, since such games can serve as a collaborative storytelling environment. The human would execute an action in the form of natural language, which would then be converted into an event representation. The event would then be fed to the *event2event* generation Sequence2Sequence [16] model, resulting in a distribution over the next possible events. Training such a neural network requires a story corpus from the fictional space the agent would inhabit, with each clause eventified and the entities and verbs generalized. Verb-to-event conversion is conducted via *VerbNet* [13], where each verb is associated with a higher-hierarchy verb class. Each verb class contains semantics, which are predicates that specify the exact role for each entity in the event. These semantics serve as models of commonsense knowledge,

and events not fitting into the predicates for the current and the previous verbs would not be valid. The filtered list of events is passed into a Deep Q Network, trained to maximize expected reward, to choose the next agent event. In order for the results to be more human-interpretable, the chosen event will then be translated back into natural language using a Long-Short Term Memory network. However, it was observed that the sequence-to-sequence network used tends to behave like simple language models and is likely to disregard the input event to directly generate a sentence present in the training dataset instead [17].

In order to improve upon the original event-to-sentence framework, Ammanabrolu *et al.* [17] devised a selection task for a sequence of tokens based on the given event. The goal of the task is to recover information lost due to the eventification process. An ensemble of modules is created to better complete the task, including (1) using a retrieve-and-edit model based on Hashimoto *et al.* [18], (2) filling event-specific templates, and (3) a series of sequence-to-sequence models, augmented with either Monte Carlo beam encoding, finite state machine encoders, or nothing (vanilla sequence-to-sequence). The full ensemble of modules out-performed the vanilla sequence-to-sequence model in the event-to-sentence task, producing sentences whose entities are labelled with their *WordNet* Synsets. Another slot-filler would translate these labels to the corresponding named entities if these entities exist in the state memory, or to a word belonging in the *WordNet* Synset. Although the slot filler recovers the named entities rather well, it may result in semantically incorrect sentences when it selects the word from the corresponding Synset.

The issue with eventifying sentences in general is that the process may ignore large portions of information present in the original sentence. For instance, prepositions indicating relative locations would be ignored by this version of event representation, as well as some additional adjectives used to modify either the subject or the object. Loss of information would result in inconsistent understandings of the current world states between the human and the agent, damaging the quality of the collaborative storytelling and gameplay.

Another relative weakness of eventified storytelling by converting events into sentences is the lack of linguistic variability. When it comes to genre-specific storytelling, different genres tend to have different prose styles and vocabulary preferences. Such discrepancies can be lost in translation after eventification and generalization, as multiple sentences convert to the same event, hence feeding different sentences into the pipeline may result in nearly identical unfilled sentences.

2.2 Transformer-based Neural Story Generation

GPT-2 is a large-scale transformer-based language model trained on large corpora scraped from websites such as Reddit [1]. It has been known to produce grammatically coherent text, yet its generated contents suffer from *logical inconsistencies and repetition* [19]. To improve the logicality of the GPT-2 model, Mao *et al.* [20] introduced commonsense knowledge into GPT-2 by encouraging the model to assign higher probabilities to sensible text options in multiple-choice format datasets such as the SWAG dataset [21]. Such an increase in probability is represented by a decrease in model perplexity on a held-out test set. Despite achieving a more than 50% decrease for both sub-word and word level perplexities after the multi-tasking fine-tuning, the focus of their work remains on generating sensible individual sentences instead of generating logically coherent paragraphs.

Guan *et al.* [22] utilized a similar approach of multi-task learning to promote logically coherent story generation in GPT-2; they implicitly introduced commonsense knowledge via pre-training on the ROCStories [23] corpus and then post-training on natural language sentences converted from commonsense knowledge triples from ConceptNet [24] and ATOMIC [25]. During post-training, the loss is a weighted sum of the cross-entropy loss from language modeling and the loss from a classification task. The goal of the classification task is for the GPT-2 model to distinguish stories that obey commonsense knowledge from stories that are logically inconsistent. Sets of negative story examples were constructed via sentence shuffling, sentence replacement, and random repetition of sen-

tences. The resulting model produces relatively sensible and coherent stories compared to multiple state-of-the-art baselines for both automatic evaluations and manual human evaluations. However, the model still produces errors such as sentence repetition, unrelated sentences or events, logical conflicts, and difficult-to-understand scenes.

AI Dungeon [1] is a text adventure game powered by the GPT-2. Players input natural language actions into the game, and the backing language model would generate descriptions of the consequences of the action. Although AI Dungeon is a relatively enjoyable text adventure game interface, its primary focus is on interactive storytelling; at every turn, the model's output does not exceed a couple of sentences. Our focus for this work, instead of improving interactive storytelling, is automatic generation of stories without extensive user guidance or input.

Although GPT-2 as a language model often successfully learns linguistic characteristics of corpora, it is not an ideal story generator as it can generate repetitive and logically incoherent content. In spite of previous efforts to incorporate better reasoning into GPT-2, there still exists logical incoherence in the generated texts. Therefore, a more rigorous logical filtering system would be necessary in order to minimize such errors.

[1] https://play.aidungeon.io/main/landing

CHAPTER 3

METHODOLOGY

We aim to simplify the automated story generation pipeline by combining the GPT-2 [1] model with the *VerbNet* [13] causal rule filter from Martin, Sood, and Riedl [12], while maintaining or improving upon the quality of the story.

3.1 Preparing the Base Model

3.1.1 Dataset

Language models statistically model the corpora that these models are trained on, so that they produce outputs that resemble entries in the training datasets. GPT-2 was originally trained on texts scraped from the internet and social media [1]. It can easily be fine-tuned given a set of texts obtained from a specific domain of interest so that the new, fine-tuned GPT-2 model would produce texts representing this domain. In our case, we would like GPT-2 model to generate plot points in a story, thus the training set also needs to contain plot points.

The Sci-Fi story dataset [17] is comprised of crowd-sourced plot summary sentences scraped from fan wikis of 11 different television shows. There are 2,276 stories in the dataset. Each story corresponds to an episode in the shows, consisting of, on average, 89.23 sentences per story. However, not all these sentences are relevant to the plot. For example,

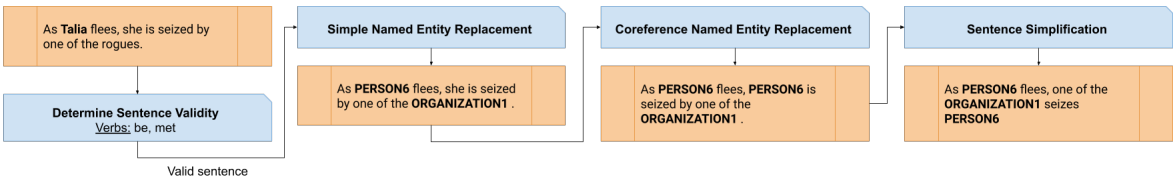

Figure 3.1: An example of how a sentence from the science fiction corpus can be simplified using the generalization process.

we can often observe sentences like "Both officers are clearly unnerved by what this bodes for the future" serving as set-ups for future plots or non-consequential continuations of the current plot. These sentences are not necessary for an action-driven story. For the purposes of this study, we are pruning these sentences out of the corpus.

3.1.2 Generalizing the Sentences

Since we aim to feed the generation results from the GPT-2 model to a parser for *VerbNet*, the fine-tuned GPT-2 model should output sentences easily parsable by *VerbNet*. The desirable characteristics of the generated sentences are: (1) has a simple syntactic structure, as in *Subject-Verb-Object*; (2) contains an active verb that is not "be", so that the sentence propels the plot; (3) does not have any specific named entity so that the logic learned through these sentences can be applied to arbitrary stories.

Based on these desirable characteristics, we pre-process the sentences from the dataset before fine-tuning GPT-2 with these sentences. The goal of this procedure is to generalize the sentences into a parseable format. In this generalization process, we utilize a list of named entities from Ammanabrolu *et al.* [17], including names of individuals, locations, organizations, specific science-fiction-related objects, or vehicles. Our specific process of generalizing sentences is detailed below and in Figure Figure 3.1:

1. **Determining Sentence Validity:** Parse the sentence through the Stanford CoreNLP parser [1] to obtain the verbs in a sentence. If the only verb (including all the different tenses) in the sentence is "be", this would indicate that the sentence is not contributing to the plot; therefore, we would discard this sentence for training purposes.

2. **Simple Named Entity Replacement:** Reference the list of named entities to find all the representations of the named entities, and replace these representations with their corresponding named entity labels. These labels are concatenated with an index indicating how many entities of the same type have occurred before this representation.

[1] https://stanfordnlp.github.io/CoreNLP/

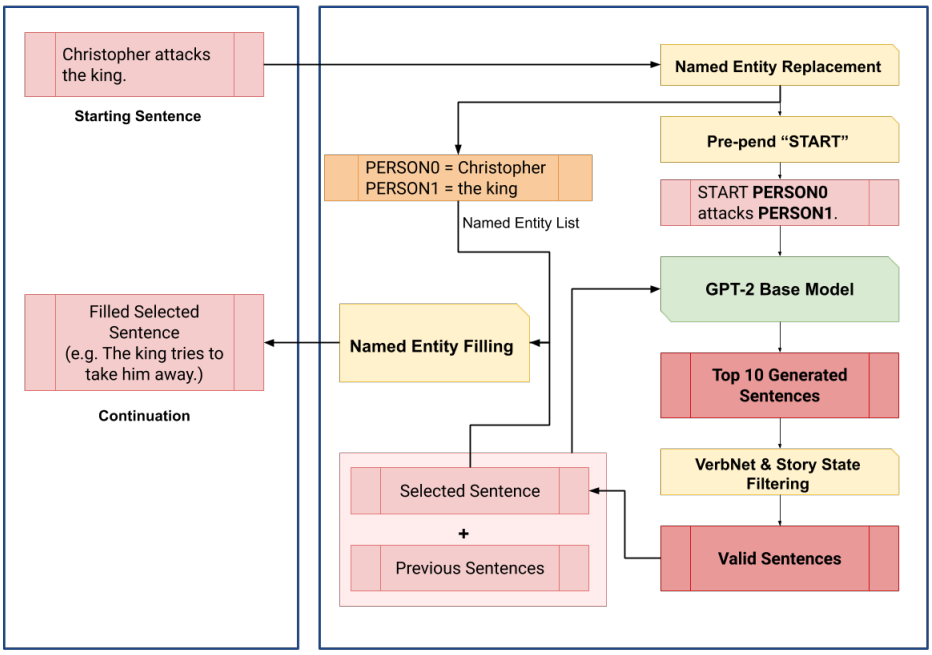

Figure 3.2: The procedure for generating a story using our pipeline.

For instance, the sentence "Takashima declares Babylon 5 open for business" would be transformed into "PERSON1 declares LOCATION0 open for business".

3. **Coreference Named Entity Replacement:** In addition to replacing the representations of the named entities (e.g. both "Scully" and "Agent Scully" are direct references to "Agent Dana Scully"), we replace co-referenced representations with the same labels as well. The co-reference clusters are inferred via `neuralcoref`. [2] With this operation, we would be changing "PERSON2 is confused as to what happened, especially when he sees the broach" to "PERSON2 is confused as to what happened, especially when PERSON2 sees the broach".

4. **Simplifying the Sentence:** We then aim to convert the sentence into a simple sentence with structure *Subject-Verb-Object*. This can be achieved by (1) converting *Subject-be-Verb(ing)* to *Subject-Verb*; (2) converting *Object-be-Verb(ed)-by-Subject* to *Subject-Verb-Object*.

[2] https://github.com/huggingface/neuralcoref

Notice that we keep a separate set of indices for every story, so that the logic and relationships between all the named entities within individual stories would stay consistent after generalization. The pre-processing also enables GPT-2 to learn generic logic underlying the different actions of the same character by replacing identifying information of the named entities with generalized labels.

After performing these operations on every sentence from the Sci-Fi dataset, we obtain a set of stories with simple, easily-parsable sentences. For each of the individual stories, we prepend the first sentence with the "START" token to indicate it as the starting of a story. We then train the GPT-2 model on this set of data sequentially to create a model that generates action-driven story plot points.

3.2 Generating the Story

Even with the model generating simple sentences representing science fiction plot points, there is still no guarantee that these outputs combine to form logically sound stories. Therefore, a filtering mechanism is still required.

In Martin, Sood, and Riedl [12], stories are created by converting a sequence of events generated from a *Seq2Seq* model into a series of sentences. The logical coherency of the events is maintained by converting each event into its corresponding *VerbNet* frame and pruning the events whose corresponding frames fail to fulfill certain verb-related constraints. For instance, the verb "kill" has a constraint that its object has to be "alive"; therefore, an event containing the verb "kill" and an inanimate object would be filtered out, and a new event would be generated based on the history of events.

For the new framework, we completely disregard eventification and map from natural language directly to *VerbNet* frames and predicates. The framework generates entire stories from a seed sentence provided by the user. At each time step, the GPT-2 model generates a set of most likely sentences using beam search to provide a continuation of the current story. Each candidate sentence is fed into a parser, which extracts and fills the *VerbNet*

frames. If a frame is found and the predicates associated with the frame are valid given the current story state, the pipeline would update the story states and proceed to the next sentence generation.

3.2.1 Starting the Story

The starting sentence of the story can be either provided by the user or from a list of pre-selected sentences from the Sci-Fi corpus. For our experiments, we are using pre-selected sentences. If the user-provided sentence only contains "be" as the verb, then the sentence would be marked as invalid and the user prompted to input another sentence. The first sentence would first have its named entity extracted and replaced by a label (see section "Generalizing the Sentences"), then the named entities and their corresponding labels would be stored in a dictionary created to help filling the labels with their named entities later.

3.2.2 Generating the Sentences

Using beam search, the trained GPT-2 model would output the top-10 continuations based on sentence history. The generated sentences would only have labels with types and indices representing the named entities in the sentence.

3.2.3 Converting to *VerbNet* Frames

We construct a parser for filling *VerbNet* frames using the Stanford CoreNLP parser [15]. First, we parse the sentence through the CoreNLP parser and obtain the part-of-speech clusters present in the parsing tree. Each cluster consists of its part-of-speech label and the words that belong in the cluster. Since our approach centers around the verbs in a sentence, we construct one verb cluster for every distinct verb in the sentence. The Stanford Dependency parser [15] provides a semantic dependency tree for words in the sentence. Using this dependency tree, we can then group the part-of-speech clusters into their corresponding

verb clusters by assigning the part-of-speech cluster to a verb if a word within the part-of-speech cluster is in a dependency relationship with the verb. These part-of-speech labels would then be matched with all of the frames from the *VerbNet* class containing the verb of the sentence. The parser finds the best-matching frame using a modified version of Levenshtein distance to prioritize *VerbNet* frames that sequentially match better with the original frame. Afterwards, the found frame would be filled with words from the part-of-speech clusters in the verb cluster.

3.2.4 Checking Frame Validity

If no frame is found in the previous step, then we discard the sentence altogether. Otherwise, we will determine the validity of the *VerbNet* frame similar to Martin, Sood, and Riedl [12]. If the frame is not valid or if the content of the sentence contradicts the current state of the story, then we prune out the sentence.

3.2.5 Outputting the Sentence

After filtering through the list of sentences, we are left with all the valid continuations. The pipeline would then select one sentence randomly from all the valid candidates. The story state memory would be updated accordingly based on what event is described by the chosen continuation. Finally, the labels in the generated sentence would again be replaced by the original named entities, and the pipeline would then output the sentence for the users to view.

3.3 Experiments

We conducted a Mechanical Turk experiment where we asked 35 Mechanical Turk workers to rate stories generated with our filtering pipeline and without. The language models used for both conditions are the same language model fine-tuned on the generalized Sci-Fi stories. An example of stories generated using both approaches is displayed in Table

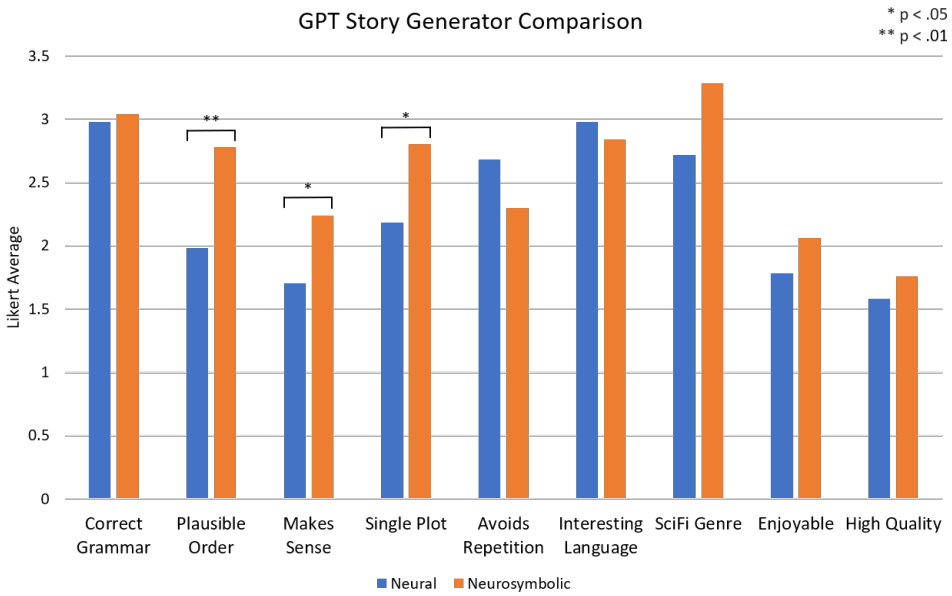

Figure 3.3: Results collected from the Mechanical Turk experiment. The "neural" statistics refer to stories generated with the unfiltered language model, while the "neurosymbolic" statistics refer to stories generated with the filtered language model. The numerical rating is the average of the Likert scale ratings, where "strongly disagree" corresponds to a score of 1 and "strongly agree" corresponds to a score of 5.

Table 3.1. We generated stories from 40 seed sentences using both approaches, resulting in 80 stories total. Five stories for each method is randomly selected for the experiments.

Each Mechanical Turk worker is presented with two stories, one story generated per approach. For every story, the worker first summarizes it to the best of their ability, then asked to rate the story on a 5-item Likert scale with the following criteria:

1. This story exhibits CORRECT GRAMMAR.
2. This story's events occur in a PLAUSIBLE ORDER.
3. This story's sentences MAKE SENSE given sentences before and after them.
4. This story FOLLOWS A SINGLE PLOT.
5. This story AVOIDS REPETITION.
6. This story uses INTERESTING LANGUAGE.

7. This story REMINDS ME OF A SPACE OPERA.

8. This story is ENJOYABLE.

9. This story is of HIGH QUALITY.

Finally, the worker is asked to identify one of the aforementioned attributes as the most important in determining the overall quality of the story and explain why.

The results of this preliminary experiment are charted in Figure 3.3. It is clear that workers perceive the filtered stories to: (1) have events occurring in more plausible orders; (2) make better sense; (3) adhere more strongly to a single plot; (4) be more similar to a space opera, which is our metrics for adhering to the science fiction genre; (5) be more enjoyable; (6) be of a higher quality. The differences in terms of genre, enjoyability, and quality are relatively less significant.

Filled Story, Filtered	**Unfilled Story, Filtered**	**Filled Story, Unfiltered**	**Unfilled Story, Unfiltered**
Anakin Solo is now eleven – and headed for the Jedi academy.	PERSON0 is DATE0 – and headed for the MISC0 academy.	Anakin Solo is now eleven – and headed for the Jedi academy.	PERSON0 is DATE0 – and headed for the MISC0 academy.
Anakin Solo is on Risa Pears's body, waiting for a Nine Two One Eight Black to appear.	PERSON0 is on PERSON2's body, waiting for a ORGANIZATION1 to appear.	Katharsis, Kassia and Griffiths discuss the possibility of the Trail Bike's escape.	LOCATION0, PERSON1 and PERSON2 discuss the possibility of the VEHICLE0's escape.
Anakin Solo's body begins to glow.	PERSON0's body begins to glow.	Mclaren Natatorium and Neelix Regretfully are convinced that it is not a Dogsled but rather a Dogsled that has been left behind, a Dogsled that is Now eleven	LOCATION1 and PERSON2 are convinced that it is not a VEHICLE1 but rather a VEHICLE1 that has been left behind, a VEHICLE1 that is DATE0
Anakin Solo's body starts to glow again, but the lights go out and the lights turn back on.	PERSON0's body starts to glow again, but the lights go out and the lights turn back on.	Dead.	Dead.
Suddenly, Anakin Solo falls unconscious.	Suddenly, PERSON0 falls unconscious.	However, Anakin Solo thinks the Arias Expedition's story is a lie and that the Submarine has no physical evidence.	However, PERSON0 thinks the ORGANIZATION1's story is a lie and that the VEHICLE0 has no physical evidence.
Anakin Solo's body is in the dark, and Vuurok sees Anakin Solo's body lying on a bed in the Vizsla of the dead Ketsu Regretfully.	PERSON0's body is in the dark, and PERSON2 sees PERSON0's body lying on a bed in the LOCATION2 of the dead PERSON3.	Peaceful Slumbers Funeral Parlor, who has never seen any Automobile in his life and never heard of anything like Anakin Solo, believes it is a fake.	LOCATION1, who has never seen any VEHICLE2 in his life and never heard of anything like PERSON0, believes it is a fake.

Table 3.1: An example of generated stories from a seed sentence in the Sci-Fi corpus. "Filtered" indicates that the story is generated with the fine-tuned GPT-2 model and the filtering mechanism, while "unfiltered" indicates that the story is generated directly from the fine-tuned GPT-2 model.

CHAPTER 4
DISCUSSION

Maintaining logical consistency throughout the stories is crucial in generating high-quality narratives. Previous works concentrated on utilizing abstractions of stories, often referred to as "events", to enforce causal relationships between different parts of the stories. When using such abstractions, a sequence of such events would be generated first, and then converted into natural language sentences. This might result in stories that, while logical, do not contain very natural-sounding language. To combat this issue, we introduce OpenAI's GPT-2 into the pipeline. GPT-2 is known to successfully mimic natural language, but has the tendency to become incoherent and lose its logical consistency. To steer GPT-2 towards generating more logically coherent stories, we use *VerbNet* to filter the generated sentences to prune out illogical continuations. The filtering is based on the established *VerbNet* logic in addition to the current story states. Our solution improves upon the previous eventified story generation models in terms of language naturalness and novelty.

There are aspects of our pipeline that can still be improved upon, which can be the emphases of future work. Currently, the GPT-2 model generates the most logical continuation based on the domain of the training corpus, but sometimes fails to produce the most logical next action for a specific character. Since all named entities in the story corpus are generalized to help GPT-2 learn the fundamental logic in storytelling, we do not introduce any characterization into our pipeline. We also do not consider how personality, social status, or previous actions by other characters effect the behaviors of characters in stories. One might be able achieve better characterization and potentially character interaction in stories by introducing special tokens or markings into the training corpus, or produce separate storylines for the different characters. Finally, the parser converting generated sentences into VerbNet frames can often fail at processing complex sentences, resulting in incorrect story

states which would affect the filtering process. Therefore, the parser will continue to be improved in the future.

Due to the complex nature of science-fiction stories, there are often multiple sub-plots co-occurring and interacting, each featuring a different set of characters. Therefore, GPT-2 trained on such data can abruptly introduce new characters without sufficient exposition. This can potentially be mitigated by either training on a different corpus with simpler story structures or generating character expositions during run time based on the story state.

CHAPTER 5
CONCLUSION

We demonstrate how symbolic semantic filtering using *VerbNet* can improve the logical consistency of stories generated by large-scale, transformer-based language models such as GPT-2. We insert the language model into our originally event-driven story generation pipeline to eliminate the event generation step in the process. Preliminary experiments illustrate that our approach improves generated stories in dimensions such as grammar, plausibility in order of events occurring in the story, and adherence to a single plot. We also detail how to prepare such a language model to be plugged into our pipeline so that our approach can be generalized for any story corpus.

The paper provides another pathway to logical story generation: a filtering mechanism applied to language model generations without explicit planning or explicit commonsense inferences. Using our method, only training a language model using a generalized story corpus would be needed to begin generating stories specific to the domain of that corpus, making our approach simple yet effective. Even when such a generalized corpus is not readily available, one can pre-process the corpus by leveraging named entity recognition tools.

REFERENCES

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019.

[2] J. R. Meehan, "The metanovel: Writing stories by computer.," YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE, Tech. Rep., 1976.

[3] M. Lebowitz, "Planning stories," in *Proceedings of the 9th annual conference of the cognitive science society*, 1987, pp. 234–242.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[5] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. of NAACL*, 2018.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. arXiv: 1810.04805.

[7] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending against neural fake news," *CoRR*, vol. abs/1905.12616, 2019. arXiv: 1905.12616.

[8] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "Ctrl: A conditional transformer language model for controllable generation," *arXiv preprint arXiv:1909.05858*, 2019.

[9] J.-S. Lee and J. Hsiang, "Patent claim generation by fine-tuning openai GPT-2," *CoRR*, vol. abs/1907.02052, 2019. arXiv: 1907.02052.

[10] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, "Dialogpt: Large-scale generative pre-training for conversational response generation," *arXiv preprint arXiv:1911.00536*, 2019.

[11] L. J. Martin, P. Ammanabrolu, X. Wang, W. Hancock, S. Singh, B. Harrison, and M. O. Riedl, "Event representations for automated story generation with deep neural nets," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] L. J. Martin, S. Sood, and M. Riedl, "Dungeons and dqns: Toward reinforcement learning agents that play tabletop roleplaying games.," in *INTWICED@ AIIDE*, 2018.

[13] K. K. Schuler, "Verbnet: A broad-coverage, comprehensive verb lexicon," 2005.

[14] C. Fellbaum, "Wordnet," *The encyclopedia of applied linguistics*, 2012.

[15] P. Qi, T. Dozat, Y. Zhang, and C. D. Manning, "Universal dependency parsing from scratch," in *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 160–170.

[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[17] P. Ammanabrolu, E. Tien, W. Cheung, Z. Luo, W. Ma, L. J. Martin, and M. O. Riedl, "Story realization: Expanding plot events into sentences," *arXiv preprint arXiv:1909.03480*, 2019.

[18] T. B. Hashimoto, K. Guu, Y. Oren, and P. S. Liang, "A retrieve-and-edit framework for predicting structured outputs," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 052–10 062.

[19] A. See, A. Pappu, R. Saxena, A. Yerukola, and C. D. Manning, "Do massively pretrained language models make better storytellers?" *arXiv preprint arXiv:1909.10705*, 2019.

[20] H. H. Mao, B. P. Majumder, J. McAuley, and G. W. Cottrell, "Improving neural story generation by targeted common sense grounding," *arXiv preprint arXiv:1908.09451*, 2019.

[21] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, "Swag: A large-scale adversarial dataset for grounded commonsense inference," *arXiv preprint arXiv:1808.05326*, 2018.

[22] J. Guan, F. Huang, Z. Zhao, X. Zhu, and M. Huang, "A knowledge-enhanced pretraining model for commonsense story generation," *arXiv preprint arXiv:2001.05139*, 2020.

[23] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen, "A corpus and evaluation framework for deeper understanding of commonsense stories," *arXiv preprint arXiv:1604.01696*, 2016.

[24] R. Speer and C. Havasi, "Representing general relational knowledge in conceptnet 5.," in *LREC*, 2012, pp. 3679–3686.

[25] M. Sap, R. Le Bras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N. A. Smith, and Y. Choi, "Atomic: An atlas of machine commonsense for if-then reasoning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3027–3035.